JURITEK: Jurnal Ilmiah Teknik Mesin, Elektro dan Komputer

E-ISSN: 2809-0799
P-ISSN: 2809-0802

Implementasi Penggunaan Teknik Keamanan Data *Hashing* Dan *Limit Login* Pada Login Sistem Pengelolaan Pencatatan Barang Dppesdm Berbasis Web

Syarhan Azmi[1*], M. Miko Sahputra Sembiring[2], Vima Zikra Adha Lubis[3], Ilhamuddin[4], dan Mhd. Furqan[5]

[1] Ilmu Komputer, Fakultas Sains & Teknologi, Universitas Islam Negeri Sumatera Utara; email : syarhanazmi07@gmail.com
[2] Ilmu Komputer, Fakultas Sains & Teknologi, Universitas Islam Negeri Sumatera Utara; email : mhdmikossbr20@gmail.com
[3] Ilmu Komputer, Fakultas Sains & Teknologi, Universitas Islam Negeri Sumatera Utara; email : vimazikraadha@gmail.com
[4] Ilmu Komputer, Fakultas Sains & Teknologi, Universitas Islam Negeri Sumatera Utara; email : ilhamuddin2424@gmail.com
[5] Ilmu Komputer, Fakultas Sains & Teknologi, Universitas Islam Negeri Sumatera Utara; email : mfurqan@uinsu.ac.id

*** Korespondensi: Syarhan Azmi**

Abstract: *This research focuses on strengthening the security architecture of the authentication module in the SIGUDANG system at the Department of Industry, Trade, Energy, and Mineral Resources (DPPESDM) of North Sumatra Province, which previously exhibited vulnerabilities to automated brute-force attacks. The primary problem identified is the risk of illegal account takeover resulting from the absence of rate limiting and robust credential protection at the database layer. The objective of this study is to develop a resilient authentication system by implementing Defense in Depth security techniques. The proposed method integrates the use of the Bcrypt hashing algorithm for password integrity and an IP-based Limit Login Attempts mechanism. Testing results using Black Box techniques with Boundary Value Analysis demonstrate that the system is capable of precisely terminating access immediately after the third failed attempt, with a blocking duration that adaptively increases from 30 seconds to 24 hours. These findings confirm that the integration of Prepared Statements, Password Hashing, and Rate Limiting effectively mitigates SQL Injection and hybrid brute-force attacks. The conclusion of this research is that the application of a multi-layered security strategy successfully reduces the probability of successful automated attacks by creating a time constraint that is computationally infeasible for attackers, thereby ensuring optimal protection of digital assets and users' personal data.*

Keywords: SIGUDANG; *Authentication Security; Bcrypt Hashing; Limit Login; Brute Force; SQL Injection; Defense in Depth; Black Box Testing*

Abstrak: Penelitian ini berfokus pada penguatan arsitektur keamanan modul autentikasi sistem SIGUDANG di DPPESDM Provinsi Sumatera Utara, yang sebelumnya memiliki kerentanan terhadap serangan *brute force* otomatis. Masalah utama yang diidentifikasi adalah risiko pengambilalihan akun secara ilegal akibat ketiadaan pembatasan laju permintaan (*rate limiting*) dan perlindungan kredensial yang kuat pada lapisan basis data. Tujuan penelitian ini adalah menciptakan sistem autentikasi yang tangguh melalui implementasi teknik keamanan berlapis (*Defense in Depth*). Metode yang diusulkan mengintegrasikan penggunaan algoritma *hashing Bcrypt* untuk integritas kata sandi dan mekanisme *Limit Login Attempts* berbasis alamat IP. Hasil pengujian menggunakan teknik *Black Box* dengan *Boundary Value Analysis* menunjukkan bahwa sistem mampu melakukan pemutusan akses secara presisi tepat setelah percobaan gagal ke-3, dengan durasi blokir yang meningkat secara adaptif dari 30 detik hingga 24 jam. Temuan ini mengonfirmasi bahwa integrasi *Prepared Statements*, *Password Hashing*, dan *Rate Limiting* efektif memitigasi serangan SQL *Injection* serta serangan *brute force* hibrida. Simpulan penelitian ini adalah penerapan strategi keamanan berlapis berhasil menurunkan probabilitas keberhasilan serangan otomatis dengan menciptakan kendala waktu yang tidak efisien bagi penyerang (*computationally infeasible*), sehingga menjamin perlindungan aset digital dan data pribadi pengguna secara optimal.

Kata kunci: SIGUDANG; Keamanan Autentikasi; *Bcrypt Hashing*; *Limit Login*; *Brute Force*; SQL *Injection*; *Defense in Depth*; *Black Box Testing*

Diterima: Oktober 20, 2025
Direvisi: Oktober 28, 2025
Diterima: Oktober 29, 2025
Diterbitkan: November 24, 2025
Versi sekarang: Januari 19, 2026

1. Pendahuluan

Seiring dengan kemajuan teknologi digital, perlindungan terhadap data pribadi dan informasi sensitif menjadi tantangan utama karena meningkatnya risiko akses ilegal dan penyalahgunaan oleh pihak yang tidak bertanggung jawab. Oleh karena itu, teknik keamanan informasi terus berkembang untuk melindungi data dari penyalahgunaan dan akses yang tidak sah. Keamanan infrastruktur komputer harus dirancang untuk melindungi aset digital dari upaya penghancuran dan eksploitasi data sensitif melalui mekanisme pertahanan yang sistematis [1].

Tingkat Serangan siber terhadap aplikasi web terus meningkat, dengan insiden pencurian kredensial dan kebocoran data pribadi (*data breach*) menjadi ancaman paling dominan. Masalah ini diperburuk oleh praktik *coding* yang mengabaikan validasi input dan mekanisme perlindungan data yang kuat [2]. Sehinga perlu adanya integritas data yang juga harus dijaga melalui pemanfaatan *Prepared Statements* di PHP untuk mencegah SQL *Injection*, memastikan *query* yang dikirimkan ke *database* selalu aman. Kebutuhan akan sistem yang tangguh ini didorong oleh prevalensi kerentanan keamanan pada lapisan aplikasi, yang seringkali dieksploitasi untuk mencuri atau merusak Data Pribadi [3].

Objek penetilian ini berfokus pada penguatan arsitektur keamanan pada modul autentikasi sistem SIGUDANG, sebuah platform pengelolaan pencatatan barang berbasis web yang diimplementasikan di Dinas Perindustrian, Perdagangan, Energi, dan Sumber Daya Mineral (DPPESDM) Provinsi Sumatera Utara. Sebagai infrastruktur digital yang mengelola inventarisasi aset daerah, SIGUDANG memiliki peran strategis dalam menjamin akuntabilitas dan transparansi pelaporan aset. Metode yang telah digunakan sebelumnya dalam pengembangan awal SIGUDANG mengadopsi model Waterfall yang berfokus pada pemenuhan fungsionalitas sistem (CRUD) dan fitur operasional seperti *Auto-Barcode* serta pelaporan otomatis [4].

Dalam aspek keamanan, sistem awal telah mengimplementasilan *Prepared Statements*, dan *Role-Based Access Control* (RBAC) untuk mengelola hak akses pengguna berdasarkan tingkatan wewenang (Admin, Staf, dan Kepala Dinas), tetapi belum mengimplementasikan mekanisme *Limit Login* untuk mencegah manipulasi permintaan lintas situs. Pengujian fungsionalitas menggunakan skenario *Use Case* pada penelitian terdahulu mengonfirmasikan bahwa seluruh fitur sistem telah berjalan secara valid. Kekuatan dari metode pengembangan sebelumnya terletak pada kemampuannya merestrukturisasi siklus pencatatan manual menjadi digital yang terintegrasi. Namun, terdapat kelemahan pada aspek ketahanan terhadap serangan tebak sandi massal (*automated brute force*). Meskipun sistem sebelumnya telah menerapkan *Prepared Statements* untuk mitigasi SQL *Injection*, fokus utama masih tertuju pada validitas fungsionalitas penggunaan sistem serta struktur dan logika kode program secara umum. Belum terdapat pemaparan detail mengenai mekanisme perlindungan kredensial pada lapisan basis data jika terjadi kompromi *server* atau strategi mitigasi untuk menghentikan bot otomatis yang melakukan percobaan login berulang. Masalah penelitian yang diidentifikasi adalah prevalensi serangan siber yang mengeksploitasi parameter login melalui pencurian kredensial dan kebocoran data pribadi (data breach). Praktik *coding* yang mengabaikan mekanisme perlindungan data yang kuat pada modul autentikasi memperburuk risiko ini. Tanpa adanya fitur pembatasan laju permintaan (*rate limiting*), penyerang dapat mengeksploitasi kecepatan komputasi untuk menebak kata sandi pengguna secara sistematis, yang berpotensi menyebabkan pengambilalihan akun secara ilegal. Sebagai solusi, penelitian ini mengusulkan implementasi teknik keamanan berlapis (*Defense in Depth*) yang mengintegrasikan penggunaan *Data Hashing* dengan algoritma *Bcrypt* dan mekanisme *Limit Login Attempts*. Pendekatan ini bertujuan untuk menciptakan sistem autentikasi yang tangguh.

2. Tinjauan Literatur

2.1. Sistem Pengelolaan Pencatatan Barang DPPESDM

Sistem didefinisikan sebagai kesatuan dari berbagai elemen dan komponen yang saling terintegrasi. Seluruh bagian dari kesatuan ini bekerja secara sinergis dan terorganisir, yang bertujuan untuk mewujudkan sasaran atau tujuan tertentu yang telah ditetapkan [5]. Sistem Pengelolaan Pencatatan Barang DPPESDM atau (SIGUDANG) adalah *platform* pengelolaan pencatatan barang berbasis web yang diimplementasikan di Dinas Perindustrian, Perdagangan, Energi, dan Sumber Daya Mineral (DPPESDM) Provinsi Sumatera Utara. Sebagai infrastruktur digital yang mengelola inventarisasi aset daerah, SIGUDANG memiliki peran strategis dalam menjamin akuntabilitas dan transparansi pelaporan aset [4].

2.2 Keamanan Sistem

Keamanan sistem merupakan elemen krusial bagi keberlangsungan mayoritas organisasi. Tantangan utama dalam merancang arsitektur pertahanan siber yang solid terletak pada kompleksitas pemetaan seluruh komponen dalam topologi sistem. Hal ini menuntut upaya ekstra dari para analis keamanan untuk memantau lalu lintas data secara menyeluruh. Fokus utama dalam pengamanan ini adalah kemampuan mendeteksi anomali: membedakan antara permintaan akses yang sah dan aktivitas yang mencurigakan, seperti lonjakan transfer data yang ekstrem atau pola koneksi perangkat lunak yang tidak wajar [6].

2.3 Algoritma Hashing

Dikembangkan oleh Niels Provos dan David Mazières pada tahun 1999, *Bcrypt* merupakan algoritma *hashing* kata sandi yang mengadopsi struktur dari *cipher* Blowfish. Sejak diperkenalkan di konferensi USENIX, protokol ini telah menjadi standar enkripsi lintas platform yang memungkinkan akses data secara konsisten di berbagai arsitektur prosesor dan sistem operasi. Secara teknis, *Bcrypt* melakukan proses *hashing* internal ke dalam kunci sebesar 448-bit, dengan ketentuan input karakter yang berkisar antara 8 hingga 56 unit [7].

2.4 *Limit Login Attempts*

Risiko keberhasilan serangan *brute-force* dapat diminimalisir dengan menerapkan batasan pada frekuensi percobaan masuk ke *dashboard* WordPress. Dalam skenario serangan ini, peretas berupaya menembus keamanan dengan menguji berbagai kombinasi *username* dan kata sandi secara berulang hingga menemukan yang tepat. Salah satu cara paling praktis untuk memproteksi situs dari ancaman ini adalah dengan menginstal plugin '*Limit Login Attempts'*, yang secara otomatis akan memblokir akses setelah jumlah kegagalan masuk melampaui batas yang ditentukan [8].

2.5 Ancaman Siber

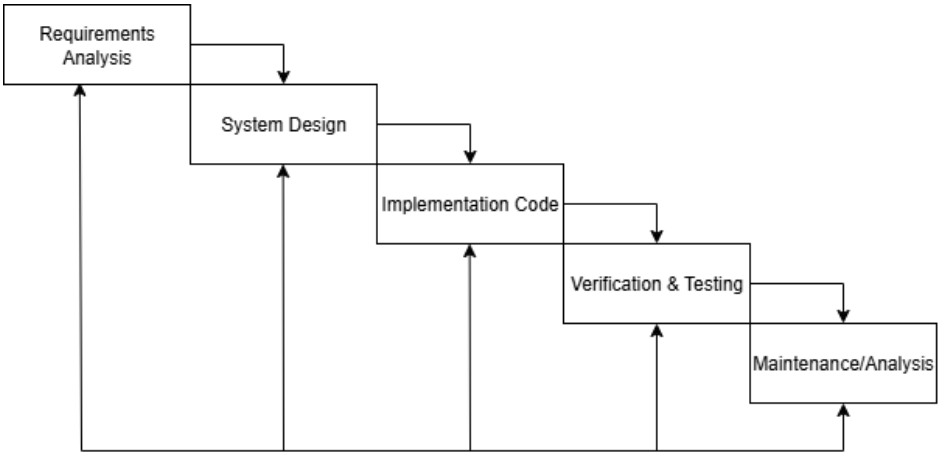
Risiko keamanan siber berakar pada dua hal: potensi serangan (ancaman) dan kondisi sistem yang rapuh (kerentanan). Ancaman adalah inisiatif jahat untuk membobol pertahanan guna merusak aset atau mengambil data, sedangkan kerentanan adalah titik lemah pada jaringan atau prosedur yang memungkinkan pembobolan itu terjadi. Risiko menjadi nyata ketika sebuah ancaman sukses mendayagunakan celah keamanan yang tersedia [9]. Siber kini bukan lagi sekadar masalah *bug* atau *glitch* teknis, melainkan senjata yang dapat melumpuhkan operasional institusi dan mengancam keamanan negara. Spektrum targetnya sangat luas, mencakup area vital seperti kesehatan, pemerintahan, dan utilitas publik. Kita melihat bagaimana *Ransomware* memaksa korban membayar tebusan besar, atau bagaimana DDoS memutus layanan *online* yang menyebabkan hilangnya kepercayaan masyarakat. Ini menunjukkan bahwa serangan siber telah bermetamorfosis menjadi ancaman serius yang merusak reputasi dan ekonomi secara masif [10].

2.6 SQL *Injection*

Structured Query Language (SQL) Injection merupakan metode manipulasi kode yang bertujuan untuk mengeksploitasi basis data melalui *input* yang tidak divalidasi. Dengan menyisipkan perintah SQL berbahaya ke dalam formulir *input*, peretas dapat menginstruksikan *server* untuk mengungkap informasi rahasia atau menghapus data penting. Serangan ini sangat berisiko karena pihak yang tidak berwenang dapat meningkatkan hak akses mereka menjadi administrator, memanipulasi catatan keuangan, hingga menghancurkan seluruh aset data yang tersimpan di dalam *server* [11].

2.7 Brute Force

Serangan *brute-force* beroperasi melalui metodologi *trial and error* guna memecahkan kode sandi pengguna. Proses ini sering kali diawali dengan tahap pengumpulan data personal target, seperti identitas lengkap atau informasi latar belakang lainnya, untuk menyusun daftar kata sandi potensial. Durasi serangan ini sangat variatif, mulai dari hitungan jam hingga bertahun-tahun bergantung pada kekuatan enkripsi yang diterapkan (misalnya 128 atau 168-bit). Terdapat beberapa varian dalam teknik ini: *Dictionary Attack* yang menggunakan daftar kata bermakna dari kamus, serta *Hybrid Brute-Force Attack* yang mengombinasikan kata kamus dengan variasi karakter tertentu. Efektivitas serangan ini sangat ditentukan oleh kompleksitas sandi, semakin acak kombinasi huruf, angka, dan simbol yang digunakan, semakin sulit bagi penyerang untuk menembus sistem autentikasi tersebut [12].



3. Metode

Proses rekayasa perangkat lunak dalam penelitian ini mengadopsi pendekatan sistematis dan sekuensial berbasis model *Waterfall*. Secara spesifik, metode yang diterapkan adalah *Iterative Waterfall Model*, yang merupakan varian evolusioner dari model klasik untuk mengakomodasi perbaikan berkelanjutan pada setiap fasenya [13].

Gambar 1. Model *Waterfall*

1. Tahap *Requirements Analysis*

Pada tahap awal ini, dilakukannya analisis terhadap vektor ancaman yang paling kritis pada sistem otentikasi web. Maka, dengan itu, ditetapkannya dua kebutuhan keamanan fungsional yang harus dipenuhi sistem usulan, yaitu:

a. Kebutuhan Integritas Data: Sistem harus mampu mencegah manipulasi *query* (SQL *Injection*) dan mengamankan penyimpanan kata sandi
b. Kebutuhan Ketersediaan: Sistem harus memiliki mekanisme pembatasan (*throttling*) untuk mencegah serangan otomatis (*Brute Force*).

2. Tahap *System Design*

Tahap ini berfokus pada transformasi kebutuhan menjadi rancangan arsitektur keamanan dan struktur data. Proses perancangan penelitian ini, dibagi menjadi dua aspek utama, yaitu:

a. Perancangan basis data dilakukan dengan menyusun skema tabel *users* untuk penyimpanan kredensial terenkripsi dan tabel login_logs guna merekam jejak audit kegagalan autentikasi.
b. Perancangan logika sistem dirumuskan melalui algoritma verifikasi yang memvisualkan skenario eksperimental, yaitu: skenario form Login, yang mengimplementasikan verifikasi berganda meliputi *Rate Limiting*, serta *Password Hashing*.

3. Tahap *Implementation Code*

Rancangan sistem diterjemahkan ke dalam bahasa pemrograman PHP dengan menggunakan editor Visual Code di lingkungan *server* lokal Laragon.

a. Pengkodean Modul Login_logs: Membuat file index. php dengan menerapkan fungsi keamanan password_hash(), password_verify(), dan session_regenerate_id.

4. Tahap *Verification & Testing*

Pengujian perangkat lunak menerapkan teknik *Black Box* untuk mengidentifikasi kesalahan fungsi, antarmuka, dan kinerja sistem. Pendekatan ini efektif untuk memverifikasi bahwa logika validasi pada formulir *login* dan manipulasi data berjalan sesuai dengan skenario yang dirancang [14]. Evaluasi keandalan sistem dilakukan melalui pengujian *Black Box* dengan teknik *Boundary Value Analysis*. Metode ini dipilih untuk menguji batasan nilai input pada formulir aplikasi, memastikan sistem mampu menangani data valid maupun tidak valid dengan respons yang tepat [15]. Pengujian dibagi menjadi dua skenario serangan:

a. Uji SQL *Injection*: Menggunakan teknik *Tautology* untuk membobol akses.

b. Uji *Brute Force*: Menjalankan skrip otomatis (bot_penyerang.php) untuk mensimulasikan serangan tebak sandi massal.

5 Tahap *Maintenance/Analysis*

Data hasil dari seluruh pengujian dikumpulkan dan dianalisis secara komparatif. Pada tahap ini, akan menampilkan segala efektivitas pertahanan sistem login_logs, yang kemudian nantinya akan dirangkum dalam kesimpulan akhir mengenai efektivitas penerapan keamanan berlapis (*Defense in Depth*) pada aplikasi web.

3.1. Studi Literatur

Afifah dan Martoyo menjelaskan bahwa studi literatur atau studi pustaka bertujuan untuk mencari berbagai teori-teori yang relevan dengan permasalahan yang sedang diteliti sebagai bahan rujukan dalam pembahasan hasil penelitian [16].

Pengembangan sistem SIGUDANG pada awalnya difokuskan pada digitalisasi manajemen aset di DPPESDM Provinsi Sumatera Utara untuk meningkatkan efisiensi operasional dan akuntabilitas pelaporan. Namun, seiring meningkatnya ancaman siber, diperlukan implementasi strategi *Defense in Depth* guna melindungi kredensial pengguna dari serangan *automated brute force* dan *SQL Injection*. Penggunaan algoritma *hashing Bcrypt* menjadi standar utama karena sifatnya yang *irreversible* dan memiliki resistensi tinggi terhadap upaya dekripsi massal dibandingkan metode konvensional. Selain pengamanan data, integritas akses dijaga melalui mekanisme *Limit Login Attempts* yang membatasi laju permintaan berdasarkan alamat IP untuk menghentikan bot otomatis secara signifikan. Integrasi ketiga pilar keamanan ini. Integritas Data melalui *Bcrypt*, Keamanan Sesi dan Pembatasan Akses, terbukti mampu menciptakan sistem autentikasi yang tangguh dan adaptif terhadap ancaman siber modern.

3.2. Metode Pengumpulan Data

3.2.1 Studi Pustaka (*Literature Review*)

Furqan dan Susilo menekankan bahwa *literature review* berfungsi sebagai instrumen strategis bagi peneliti untuk memetakan urgensi masalah dan arah riset yang akan dilakukan. Melalui penelusuran terhadap studi terdahulu, peneliti mampu membedah cakupan eksplorasi yang sudah ada sekaligus menemukan celah atau keterbatasan metodologis yang perlu diperdalam lebih lanjut [17]. Metode ini dilakukan dengan mengumpulkan dan menganalisis referensi teoritis dari buku, jurnal ilmiah terakreditasi, dan standar keamanan internasional. Fokus studi pustaka meliputi: Teori dasar mengenai algoritma kriptografi *Hashing* (*Bcrypt*). Mekanisme kerja manajemen sesi (*Session Management*). Pola serangan *Brute Force* dan teknik mitigasi *Rate Limiting*. Standar keamanan kode (*Secure Coding Practices*) pada bahasa pemrograman PHP.

3.2.2 Observasi Eksperimental (*Experimental Observation*)

Penelitian ini dilakukan dengan pendekatan observasi secara langsung terhadap perilaku sistem di lingkungan (*localhost*). Data dikumpulkan melalui serangkaian percobaan terkontrol. Kerentanan pada lapisan basis data (*database layer*) masih menjadi ancaman serius bagi integritas sistem informasi. Penelitian terbaru oleh Kurniawan et al. (2025) menunjukkan bahwa celah SQL *Injection* pada parameter URL yang tidak divalidasi memungkinkan penyerang untuk mengekstraksi informasi sensitif, seperti struktur tabel dan kredensial pengguna, menggunakan alat otomatisasi. Temuan ini menegaskan urgensi penerapan mekanisme pertahanan preventif seperti *Prepared Statements* dalam pengembangan aplikasi web [18].

a. Observasi Kerentanan: Mengamati respons sistem saat menerima *input* berbahaya (seperti *payload* SQL *Injection* ' OR '1'='1).

b. Observasi Log Serangan: Memantau perubahan data pada tabel login_logs di *database* MySQL saat skrip *bot* penyerang dijalankan, untuk memverifikasi apakah sistem berhasil mendeteksi dan mencatat alamat IP penyerang.

3.2.3 Dokumentasi (*Documentation*)

Pendekatan dokumentatif merupakan prosedur perolehan data yang berfokus pada penghimpunan informasi melalui sumber-sumber tertulis maupun literatur yang tersedia. Instrumen yang digunakan dalam metode ini mencakup beragam format, mulai dari laporan resmi, catatan historis, arsip instansi, hingga jurnal ilmiah dan rekaman data lainnya yang memiliki relevansi kuat terhadap fokus penelitian [19]. Metode ini dilakukan dengan mendokumentasikan seluruh proses input dan *output* yang dihasilkan selama proses penelitian, dengan tangkapan layar (*screenshots*) sebagai bukti keberhasilan serangan pada sistem rentan dan bukti penolakan akses pada sistem aman.

3.3. Teknik Pengujian

Untuk memvalidasi ketahanan (*robustness*) modul autentikasi, dilakukan pengujian terhadap mekanisme keamanan *hashing* dan pembatasan akses (*limit login attempts*). Pengujian ini dilaksanakan menggunakan simulasi vektor serangan standar, yang meliputi serangan *SQL Injection* dan *Brute Force*. Kerentanan *Session Fixation* dan *Session Hijacking* mengancam otorisasi pasca-login dengan mencuri *token akses* pengguna, yang sering kali diawali oleh serangan *Cross-Site Scripting* (XSS) [20]. Serangan terhadap aplikasi web seringkali menargetkan celah umum seperti *SQL Injection*, *Cross Site Scripting (XSS)*, dan *Cross Site Request Forgery (CSRF)*. Kerentanan ini memungkinkan penyerang untuk mencuri data, memanipulasi basis data, atau mengganggu ketersediaan layanan, sehingga memerlukan mekanisme pertahanan berlapis pada sisi *server* [21]. Ketiadaan mekanisme pembatasan akses (*rate limiting*) pada formulir login merupakan kerentanan fatal yang sering diabaikan. Studi kasus yang dilakukan oleh Yusuf dan Suharsono (2023) membuktikan bahwa tanpa adanya *Limit Login Attempts*, penyerang dapat melakukan serangan *Brute Force* sebanyak ratusan kali (519 percobaan dalam pengujian mereka) tanpa terblokir oleh sistem. Hal ini menegaskan urgensi penerapan fitur pembatasan upaya login untuk mencegah eksploitasi otomatisasi [22]. Berikut merupakan metode serangan yang akan diuji dalam sistem keamanan login:

Tabel 1. Metode Pengujian

VEKTOR SERANGAN	METODE PENGUJIAN
SQL *Injection*	Menggunakan *Payload* Tautology (' OR '1'='1) untuk membobol otentikasi.
Brute Force	Menggunakan *script* cURL/PHP Bot (bot_penyerang.php) untuk simulasi *Dictionary Attack* yang cepat

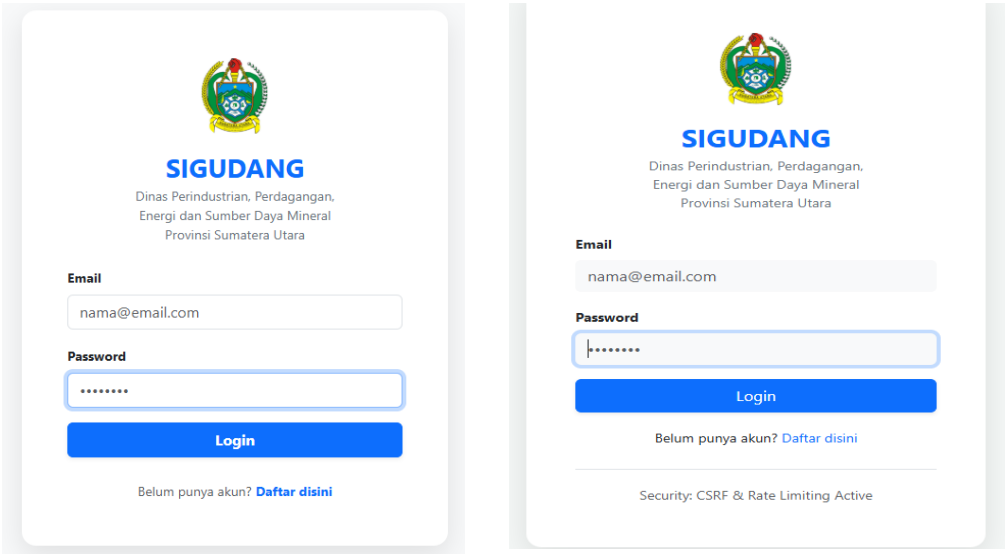
Pada Tabel. 1 Metode Pengujian, metodologi pengujian keamanan yang diterapkan mencakup simulasi penetrasi terhadap dua vektor serangan utama, yaitu SQL *Injection* dan *Brute Force*, guna mengevaluasi integritas mekanisme autentikasi sistem. Eksploitasi dilakukan melalui pendekatan teknis yang berbeda; pada aspek kerentanan basis data, pengujian menggunakan teknik tautologi untuk memanipulasi logika kueri SQL, sementara pada aspek ketahanan akses, digunakan otomasi skrip berbasis cURL atau PHP Bot untuk menjalankan *dictionary attack* secara masif. Secara keseluruhan, kedua metode ini berfungsi sebagai instrumen sederhana untuk mengidentifikasi celah pada lapisan validasi input dan efektivitas kontrol akses terhadap upaya peretasan.

4. Hasil dan Pembahasan

Pada tahap ini, dilakukan pengujian keamanan sistem (*security testing*) terhadap aplikasi inventaris berbasis web (SIGUDANG). Pengujian difokuskan pada dua vektor serangan utama yang paling umum terjadi pada sistem autentikasi, yaitu *SQL Injection* (SQLi) dan *Brute Force Attack*. Pengujian dilakukan menggunakan skrip simulasi serangan otomatis (*automated attack script*) untuk memvalidasi ketahanan sistem terhadap upaya akses ilegal.

4.1 . Implementasi Sistem Login

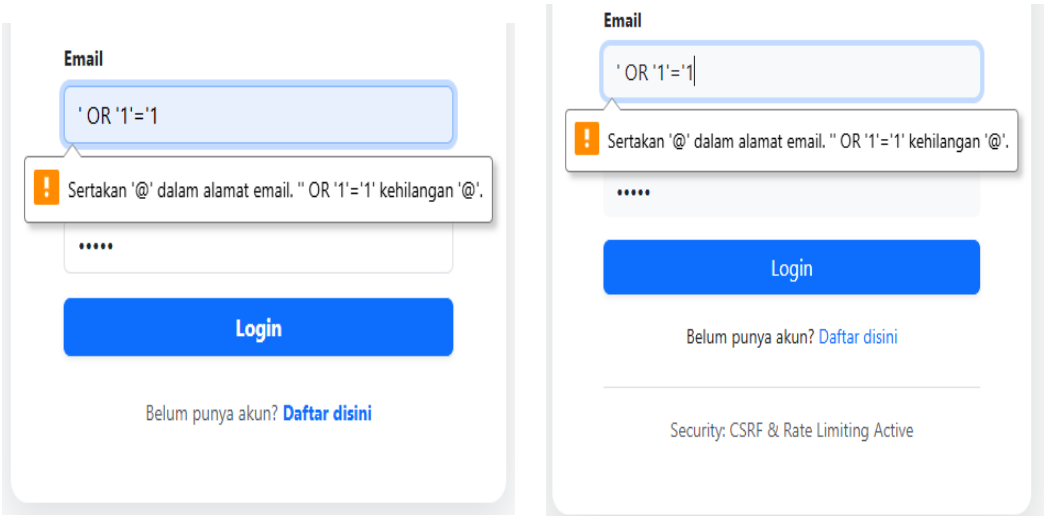
a. Halaman Login



Gambar 2. Halaman Login Sebelum (a) dan Halaman Login Sesudah (b)

Pada visualisasi **Gambar 2.** Menampilkan form login, yang dimana (a) gambar kiri, sistem login yang telah menerapkan metode *Prepared Statements* untuk mencegah serangan *SQL Injection*. Namun, antarmuka belum memberikan batasan terhadap jumlah percobaan login (*Rate Limiting*), sehingga sistem masih rentan terhadap serangan *Brute Force* yang dilakukan secara berulang-ulang. Sedangkan pada tahap pengembangan selanjutnya (b) gambar kanan, keamanan sistem ditingkatkan dengan fitur *Login Rate Limiting*. Antarmuka kini menampilkan pesan "Akses Ditangguhkan" dan mengunci *form input* apabila terdeteksi aktivitas login mencurigakan, sehingga menutup celah keamanan dari serangan *Brute Force*.

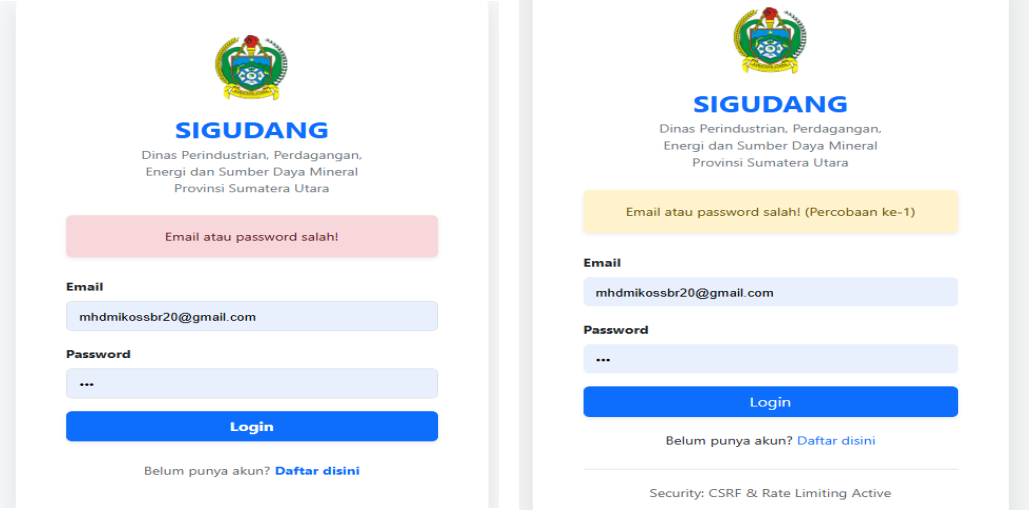
b. Pengujian Kerentanan SQL *Injection* dengan Metode Tautology



Gambar 3. Respon Kedua Sistem Login Terhadap Metode *Payload* Tautology (' OR '1'='1)

Berdasarkan hasil pengujian **Gambar. 3** menggunakan *payload Tautology* (' OR '1'='1), kedua sistem login baik (sistem login sebelum dan sesudah) terbukti mampu menolak manipulasi logika *query database*, yang seharusnya karena *FALSE* OR *TRUE* = *TRUE*. Maka, keadaan tersebut bisa memaksa *database*, menganggap seluruh permintaan ini adalah Benar. *Database* langsung memberikan data *user* pertama yang dia temukan, tanpa peduli passwordnya apa, tetapi hal itu dicegah dengan pengimplementasian *Prepared Statements*. Hal ini menunjukkan bahwa implementasi *Prepared Statements* pada kode program telah berfungsi dengan baik dalam memisahkan data *input* pengguna dari perintah SQL, sehingga integritas *database* tetap terjaga dan penyerang tidak dapat mem-*bypass* proses login tanpa kredensial yang valid.

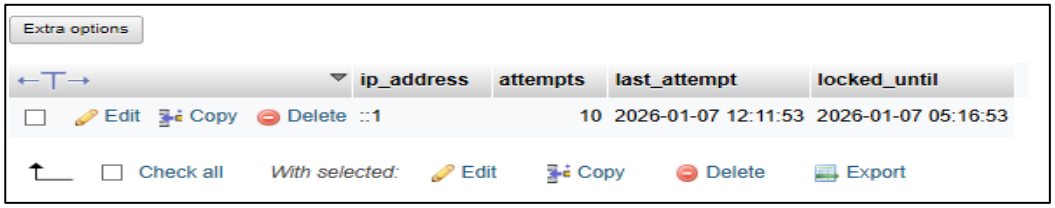
c. Uji Penerapan Mekanisme *Rate Limiting*



Gambar 4. Uji Kedua Sistem Login Dengan *Password* Yang Salah

Pada skenario **Gambar 4** sistem login yang sebelumnya (a) sudah "naik level". Selain memakai *Prepared Statements*, telah ditambahkannya logika pengecekan ke tabel login_attempts sebelum memproses login. *Update* dari sistem keamanan login (a) kiri, adalah sistem keamanan login (b) kanan, sistem b akan menghitung setiap kegagalan dari satu alamat IP. Jika kegagalan mencapai ambang batas (misal: 3 kali, 10 kali, dan 13 kali), sistem akan memutus akses (blokir) terhadap IP tersebut selama durasi tertentu (30 detik untuk 3 kali, 5 menit untuk 10 kali, dan 24 jam untuk 13 kali). Sehingga, implementasi *Rate Limiting* ini, dapat mencegat akses yang tidak sah dan serangan *brute force* yang selalu menebak-nebak sistem, dan akan menurunkan akses login dalam jangkau waktu tertentu, bahkan dalam jangka waktu yang lama, sampai email dan *password* dalam keadaan benar. Jika benar, akses akan kembali kepada limit pertama. Ini merupakan penerapan dari standar keamanan berlapis (*Defense in Depth*). Berbeda dengan sistem keamanan login sebelumnya (a) kiri, yang hanya memakai metode *Prepared Statements* saja tanpa penerapan *Rate Limiting*, yang dapat berkali-kali mencoba email dan *password* walaupun keadaannya salah dan dengan akses dalam waktu yang tanpa batas, sampai keadaannya benar. Namun, karena tanpa pembatasan akses ini, sangat rentan terhadap *Hybrid Brute-Force Attack*. Maka dari itu penggunaan *Rate Limiting* adalah hal yang sangat tepat untuk pembatasan akses. Pada sistem (b) kanan, penerapan *Rate Limiting* tidak hanya berfungsi sebagai penghambat waktu, tetapi juga memaksa validasi berbasis identitas jaringan (*IP-based Validation*). Hal ini krusial karena meskipun penyerang mencoba memanipulasi atau menghapus sesi browser (*clearing cookies*) untuk mengelabui sistem agar dianggap sebagai pengguna baru, blokir tetap berlaku karena terikat pada alamat IP.

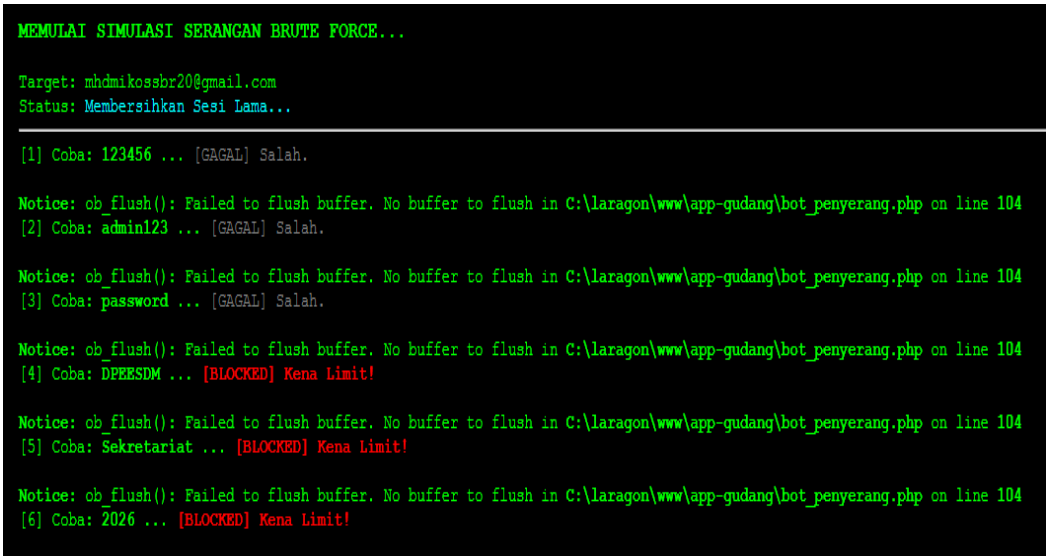
d. Analisis Forensik Pencatatan Aktivitas Login pada *Database*



Gambar 5. Log Forensik Alamat IP dan Riwayat Pelanggaran pada Tabel login_attempts

Gambar 5. menampilan tabel login_attempts di phpMyAdmin. Terlihat sistem mencatat Alamat IP (::1), jumlah percobaan sampai (attempts: 10), serta waktu penguncian (locked_until) yang aktif. Meskipun penyerang telah menggunakan bot yang mungkin menghapus *cookies*, sistem tetap mampu mengenali penyerang melalui alamat IP yang tercatat di *database*, sehingga blokir tetap berlaku secara akurat.

e. Simulasi Serangan Brute Force pada sistem login dengan Rate Limiting



Gambar 6. Simulasi Serangan *Brute Force* pada sistem login dengan *Rate Limiting*

Pada simulasi serangan **Gambar 6.** penulis menguji serangan *Brute Force* menggunakan *script* cURL/PHP Bot (bot_penyerang.php) untuk simulasi *Dictionary Attack* yang cepat. Dapat dilihat di pada gambar, ketika bot atau peretas mengumpulkan data-data dengan metode *Trial and Error*, yaitu mengumpulkan informasi berdasarkan tentang si pengguna, maka dibuatnya lah *script* peretasan yang isinya ribuan *library password* lemah yang di gabungkan. Maka, dengan adanya keamanan berlapis menggunakan *Rate Limiting* ini, akan mencegah setiap serangan *library* (tebak *password* lemah) tersebut. Maka, *password* yang tidak sesuai dengan si pengguna, akan di blokir dengan oleh *Rate Limiting*, sesuai jangka waktu dan logika login yang digunakan tadi. Yaitu (30 detik untuk 3 kali *password* gagal, 5 menit untuk 10 kali *password* gagal, dan 24 jam untuk 13 kali *password* gagal), hal ini bahkan dapat di *upgrade* kembali, sesuai keinginan yang ditentukan. Dengan teknik *Rate Limiting* ini, kecil kemungkinan sistem dapat diretas, pastinya akan membutuhkan waktu yang berbulan-bulan, berjam-jam, bahkan bertahun-tahun, untuk meretas data pengguna. Berbeda dengan sistem login (a) yang hanya mengandalkan *Prepared Statements*, yang kemungkinan sangat sensitif sekali tanpa adanya pembatasan akses yang tidak sah.

4.2 Hasil Pengujian Sistem

Pada tahap ini, pengujian sistem login SIGUDANG dilakukan menggunakan metode *Blackbox Testing* dengan teknik *Boundary Value Analysis* (BVA). Pemilihan metode ini bertujuan untuk memvalidasi fungsionalitas sistem dalam menangani *input* pengguna serta memastikan mekanisme keamanan bekerja tepat pada ambang batas (*boundary*) yang telah ditentukan.

Tabel 2. Hasil Pengujian *Boundary Value Analysis* (BVA)

ID	Fitur Yang Diuji	Nilai *Input*	Titik Batas (*Boundary*)	Respon Sistem Yang diharapkan	Hasil
BVA-01	*Password Hashing*	*Input password* benar	*Valid Match*	*Bcrypt* memverifikasi *hash* di DB, akses diterima.	Berhasil
BVA-02	*Password Hashing*	*Input password* Salah	*Invalid Match*	Hash tidak cocok, login ditolak, *counter* bertambah 1.	Berhasil
BVA-03	*Limit Login* (1)	Percobaan gagal ke-2	*Under Limit*	Form tetap aktif, muncul pesan "Percobaan ke-2".	Berhasil
BVA-04	*Limit Login* (2)	Percobaan gagal ke-3	*Exact Limit*	Sistem mengunci IP, muncul pesan "Blokir 30 Detik".	Berhasil
BVA-05	*Limit Login* (3)	Percobaan gagal ke-4	*Over Limit*	Form tertutup/dikunci meskipun *password* benar.	Berhasil
BVA-06	*Limit Login* (4)	Percobaan gagal ke-10	*Secondary Limit*	Durasi blokir meningkat otomatis menjadi 5 menit.	Berhasil

BVA-07	*Rate Limiting*	Penghapusan *Cookie* Browser	*Session Bypass*	Blokir tetap aktif karena deteksi berbasis alamat IP.	Berhasil

Berdasarkan tabel pengujian di atas (**Tabel. 2**), Pengujian menunjukkan bahwa keamanan data pengguna tidak hanya bergantung pada kerahasiaan *password* yang disimpan dalam bentuk *hash Bcrypt*, tetapi juga pada frekuensi upaya autentikasi. Teknik BVA membuktikan bahwa sistem mampu menangani kondisi *valid match* (BVA-01) dan *invalid match* (BVA-02) secara akurat sebelum data diproses oleh fungsi password_verify(). Sesuai dengan konsep *Defense in Depth*, sistem SIGUDANG menunjukkan ketahanan pada titik batas kritis (*Exact Limit*). Pada pengujian BVA-04, sistem secara presisi melakukan pemutusan akses tepat setelah percobaan ke-3. Hal ini sangat krusial untuk menghentikan serangan *Hybrid Brute-Force* sebelum penyerang memiliki kesempatan lebih luas untuk menebak kunci *hash*. Sistem menunjukkan perilaku adaptif pada pengujian BVA-06, di mana durasi blokir meningkat secara signifikan (5 menit hingga 24 jam) seiring bertambahnya frekuensi kegagalan. Analisis forensik pada tabel login_attempts (**Gambar. 5**) mengonfirmasi bahwa data IP (::1) tetap terkunci persisten meskipun dilakukan upaya manipulasi sesi atau penghapusan *cookie* (BVA-07). Secara keseluruhan, pengujian berbasis *Blackbox* dengan teknik *Boundary Value Analysis* ini memvalidasi bahwa implementasi teknik *Hashing* dan *Limit Login* pada sistem SIGUDANG telah memenuhi kriteria keamanan data yang tangguh. Sistem berhasil meminimalisir risiko pengambilalihan akun secara ilegal dengan memaksa penyerang menghadapi kendala waktu yang tidak efisien (*computationally infeasible*).

5. Kesimpulan

Penelitian ini berhasil mengimplementasikan penguatan arsitektur keamanan pada sistem autentikasi SIGUDANG melalui strategi *Defense in Depth*. Temuan utama menunjukkan bahwa penggunaan algoritma *hashing* Bcrypt memberikan perlindungan kredensial yang kuat di lapisan basis data, sementara mekanisme *Rate Limiting* efektif dalam menghentikan serangan *automated brute force*. Bukti pengujian melalui teknik *Boundary Value Analysis* (BVA) mengonfirmasi bahwa sistem mampu melakukan pemutusan akses secara presisi tepat setelah ambang batas percobaan ke-3 terlampaui, dengan durasi blokir yang meningkat secara adaptif hingga 24 jam.

Sintesis dari temuan ini menunjukkan hubungan yang selaras dengan tujuan penelitian, di mana integrasi *Prepared Statements*, *Password Hashing*, dan *Limit Login* berhasil menutup celah keamanan yang sebelumnya ada. Hal ini mendukung argumen bahwa validitas fungsionalitas sistem harus dibarengi dengan mekanisme pertahanan proaktif untuk menjaga integritas data pribadi pengguna. Implikasi dari penelitian ini memberikan kontribusi praktis bagi institusi pemerintah, khususnya DPPESDM Provinsi Sumatera Utara, dalam menjamin akuntabilitas aset melalui infrastruktur digital yang tangguh terhadap ancaman siber modern. Secara teoritis, penelitian ini memperkaya kajian mengenai penerapan *Secure Coding Practices* pada bahasa pemrograman PHP.

Meskipun telah mencapai tujuannya, penelitian ini memiliki keterbatasan pada fokus pengujian yang masih berbasis lingkungan *server* lokal (*localhost*). Oleh karena itu, saran untuk penelitian selanjutnya adalah melakukan pengujian pada lingkungan *server* produksi yang sesungguhnya serta mempertimbangkan penggunaan metode autentikasi tambahan seperti *Multi-Factor Authentication* (MFA) untuk meningkatkan lapisan perlindungan akses pengguna.

Kontribusi Penulis: Konseptualisasi: MMSS dan I; Metodologi: VZAL dan SA; Perangkat Lunak: SA; Validasi: MMSS, VZAL, I, MF, dan SA; Analisis formal: MF; Investigasi: MMSS, I, dan SA; Sumber daya: MF; Kurasi data: MMSS; Penulisan—persiapan draf asli: SA; Penulisan—peninjauan dan penyuntingan: MMSS; Visualisasi: VZAL; Supervisi: MF; Administrasi proyek: SA; Akuisisi pendanaan: I, VZAL, dan SA.

Pendanaan: Penelitian ini didanai oleh kedua Orang Tua penulis.

Pernyataan Ketersediaan Data: Kami sangat bersedia untuk semua kepenulisan dan data makalah ini dipublikasikan.

Ucapan Terima Kasih: Penulis mengucapkan terima kasih atas dukungan yang diberikan atas semua pihak, terutama kepada Ayah dan Ibu yang selalu mendoakan kami, yang membakar semangat juang kami untuk bisa merasakan barisan, belajar dibangku kuliah ini.

Ucapan terima kasih juga, kepada tim redaksi jurnal atas segala kesempatannya. Selama penulisan artikel ini. Benar, penulis penggunaan perangkat GEMINI AI hanya sebagai pedoman saja, namun segala teoritisnya tetap dari kerangka berifikir dan diskusi penulis, jadi tidak seluruhnya bergantung kepada AI.

Konflik Kepentingan: Penulis menyatakan tidak ada konflik kepentingan.

Referensi

[1] N. A. Prasetiyo, R. B. Huwae, and A. H. Jatmika, "AUDIT DAN ANALISIS WEBSITE PEMERINTAH MENGGUNAKAN PENGUJIAN PENETRASI SQL INJECTION DAN CROSS SITE SCRIPTING (XSS)," vol. 6, no. 2, pp. 525–533, 2024.

[2] M. Syukri, Imam Riadi, and Tole Sutikno, "Analisis Forensik Keamanan Data Pribadi pada Mode Privasi Browser Menggunakan Metode National Institute of Standards and Technology (NIST)," J. Process., vol. 20, no. 1, pp. 58–67, 2025, doi: 10.33998/processor.2025.20.1.2012.

[3] V. Simangunsong, Y. R. Hutasoit, D. Siallagan, T. Rekayasa, P. Lunak, and I. T. Del, "Analisis Terhadap Keamanan Password Menggunakan Hash SHA-256," J. Quancom, vol. 3, no. 1, pp. 13–17, 2025, [Online]. Available: https://doi.org/10.62375/jqc.v3i1.431

[4] C. Umri and M. M. S. Sembiring, "Perancangan Dan Implementasi Sistem Pengelolaan Pencatatan Barang DPPESDM Berbasis Web," vol. 10, no. 2, pp. 183–194, 2025.

[5] R. J. Oktafiani and F. Anggraini, "Sistem informasi inventaris barang berbasis web pada sma budi mulia utama," vol. 9, no. 2, pp. 24–36, 2023.

[6] J. T. Santoso, ARTIFICIAL INTELLIGENCE DAN DATA MINING Dalam Kerangka Sekuriti. Semarang: Yayasan Prima Agus Teknik Bekerja sama dengan Universitas Sains & Teknologi Komputer (Universitas STEKOM), 2023.

[7] S. T. Amed Leyva-Mederos, Ania Rosa Hernández Quintana, Fernando Ortiz-Rodriguez, Jose L. Martinez-Rodriguez, Semantic Web Technologies and Applications in Artificial Intelligence of Things. Pennsylvania: IGI Global, 2024.

[8] W. A.-I.-O. for Dummies, WordPress All-In-One for Dummies. New Jersey: John Wiley & Sons, Incorporated, 2024.

[9] A. E. Syaputra et al., KEAMANAN JARINGAN KOMPUTER. Banten: PENERBIT PT SADA KURNIA PUSTAKA, 2025.

[10] A. C. Jefri Andriadi, Finola Fiftem Eka Putri, INFORMASI SECURITY. Kota Payakumbuh: Penerbit Fahmi Karya, 2025.

[11] J. Kaba, MAKRIFATNYA MAKRIFAT TITIK SATU. Indramayu: CV. Adanu Abimata, 2023.

[12] S. M. R. Noval, Soecipto, and A. Jamaludin, PERLINDUNGAN HAK DIGITAL Ancaman Privasi di Tengah Serangan Social Engineering. Kota Depok: PT RAJAGRAFINDO PERSADA, 2022.

[13] M. S. Deni Murdiani, "PERBANDINGAN METODOLOGI WATERFALL DAN RAD (RAPID APPLICATION DEVELOPMENT) DALAM PENGEMBANGAN SISTEM INFORMASI," JINTEKS (Jurnal Inform. Teknol. dan Sains), vol. Vol. 4 No., no. November, p. hlm. 302 – 306, 2022.

[14] R. Putra Fajar, "Teknik Boundary Value Analysis pada Blackbox Testing untuk Aplikasi Buku Catatan Harian," J. Repos., vol. 6, no. 1, pp. 69–78, 2024, doi: 10.22219/repositor.v6i1.31852.

[15] A. Zahra et al., "Pengujian Black Box pada Website Jasa Angkutan dan Ekspedisi Menggunakan Teknik Boundary Value Analysis," Media J. Inform., vol. 16, no. 2, p. 133, 2024, doi: 10.35194/mji.v16i2.4798.

[16] M. Afifah Ansori, "Mencari Tambahan Ilmu," Pengertian J. Pendidik. Indones., vol. Vol.2, No., pp. 137–144, 2024, doi: https://doi.org/10.61930/pjpi.v2i1.

[17] S. S. Furqan, Mhd., METODOLOGI PENELITIAN. Sijunjung: PENERBIT MITRA CENDEKIA MEDIA, 2025.

[18] N. D. Kurniawan et al., "ANALISIS KERENTANAN SQL INJECTION MENGGUNAKAN," vol. 10, no. 1, 2025.

[19] Nova Christian Mamuaya; Wahyudi; Nurhasan Syah M. Zainal Arifin; Jefri Kurniawan Ahmad Herlyasa Sosro Pratama; Indri Gus Permata Sari Asmalinda Sy; Hendro Sukoco; Lisa Hermawati, Metode Penelitian Kuantitatif. Kota Padang: Azzia Karya Bersama, 2025.

[20] A. Asvin, M. Suradi, and W. Saputra, "Strategi Keamanan Router MikroTik : Deteksi dan Mitigasi Serangan Brute Force Berbasis Scripting," vol. 7, pp. 12–19, 2025.

[21] R. Rusydi and S. Arlis, "Jurnal KomtekInfo Penerapan Acunetix Vulnerability Scanner dari Serangan Siber pada Keamanan Website Kampus," vol. 11, pp. 173–180, 2024, doi: 10.35134/komtekinfo.v11i3.569.

[22] R. R. Yusuf and T. N. Suharsono, "PENGUJIAN KEAMANAN DENGAN METODE OWASP TOP 10 PADA WEBSITE EFORM HELPDESK," pp. 402–413, 2023.